



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/784,375	02/23/2004	Pal Takacsi-Nagy	BEAS-01389US1	8905
23910	7590	05/30/2008	EXAMINER	
FLIESLER MEYER LLP 650 CALIFORNIA STREET 14TH FLOOR SAN FRANCISCO, CA 94108			VO, TED T	
			ART UNIT	PAPER NUMBER
			2191	
			MAIL DATE	DELIVERY MODE
			05/30/2008	PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b> 10/784,375	<b>Applicant(s)</b> TAKACSI-NAGY, PAL	
	<b>Examiner</b> TED T. VO	<b>Art Unit</b> 2191	

**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 03 March 2008.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 50-63 and 71-75 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 50-63 and 71-75 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)                       | 4) <input type="checkbox"/> Interview Summary (PTO-413)           |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)   | Paper No(s)/Mail Date. _____                                      |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date <u>3/3/08</u> .  | 6) <input type="checkbox"/> Other: _____                          |

### **DETAILED ACTION**

1. This action is in response to the amendment filed as RCE on 03/03/2008.

Claims 64-70 are canceled; claims 71-75 are newly added.

Claims 50-63, 71-75 are pending in the application.

### ***Response to Arguments***

2. This Action is non-final to address Applicants' further request for considering the amendment and Applicant's arguments in the Remarks filed on 03/03/2008.

The Applicant's specification appears adding a workflow construct to an existing language such as Java programming language, where the workflows construct is such as in XML. It should be noted that Java might direct to trademarks, where if a trade mark or trade name appears in the specification it should be with the symbol <sup>TM</sup> or ®, accordingly.

As required under 1.111(b) and (c), the amendment should address the patentability of the claimed subject matters. The amendment to the claim appears it was as the source file, like Java Programming, stored in a computer medium, where including with the workflows definition, it is now with program source code and class, i.e. another expression for Java programming. Thus, this amendment does not fully address the patentability because Java is known as programming language for having code and classes. The amendment is further added the workflow definition is created using a workflow language, now the workflow language is specified in form of annotation to the source code and the classes. Clearly, the amendment

merely address such the workflow language is such as XML, where this markup language is known using tags and attributes for annotations, and is provided in the art for constructing workflow definitions. Accordingly, the whole specification including claiming is toward writing a program without addressing any patentability. It should be noted that writing a program is only a common way of the art, and used by anyone who is writing computer programming. It cannot address the requirements under 1.111(b) and (c).

Miller discloses how to uses workflow constructs to create workflow definitions. See Figure 1-2, p. 5, A Workflow Definition Tool generates Workflow Definition. The Workflow language comprises Java language. See sec. 2.3, p. 28. Thus, all programmers can extend a program with workflow constructs.

On the other hand, Peltz discloses, “workflow definition” using WSFL (See its “early work”, p. 5), where WSFL includes Java (See its first paragraph, p. 11), or JavaDoc annotation tags (See its p. 10).

It is obvious to the ordinary in the art at the time to combine or include Application program(s) with Java Application because Java is common developing web and business applications. Furthermore, using Java Application in the Muller’ Applications only yields the same and predictable results, and it is obvious for every ordinary in the art to include it in the teaching of Muller.

***Claim Rejections - 35 USC § 103***

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

A person shall be entitled to a patent unless –

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claims 50-63, 71-75 are rejected under 35 U.S.C. 102(a) as being unpatentable by Muller, “Event-Oriented Dynamic Adaptation of Workflows” Model, Architecture, and Implementation”, in view of Peltz, “Web Service Orchestration”, HP, 1-2003.

As per claim 50: Muller discloses, a workflow definition tool for utilizing a workflow language (i.e. application programs and its application data used as a workflow engine for run-time (p. 5, and Figure 1-2)). Muller’s tool creates a source file (i.e. a box “workflow definition”) with workflow constructs to reference to the workflow language and workflow model (See Figure 1-2), which are acting as a workflow execution engine. The workflow language in general is a procedural language such as C++ or Java (p.28), or functional languages LIPS or logic language Prolog. Muller tool execution engine is focused on Procedural Languages.

A Workflow system of Muller discloses (depicting claim 50),

Art Unit: 2191

*A system for utilizing a workflow language, comprising:*

*a computer including a processing device operating thereon* (Every computer has a CPU/memory);

*a program source file stored on a computer readable medium* (See Figure 1-2, p. 5, A

Workflow Definition Tool generates Workflow Definition), *wherein the program source file*

*includes a source code and classes therein and workflow definition created using a workflow*

*language that is specified in the form of annotations to the source code and classes, and*

*wherein said workflow language extends the source code with a plurality of workflow*

*constructs* (See p. 2, “incorporating the process logic directly into the application program”. See

Figure 1-2, p. 5, An extent of the Workflow Definition to Application Program(s): The

Application Program is invoked by Workflow Engine – The Workflow language comprises Java

language (Sec. 2.3, p. 28)), *including constructs for defining parallel processing of a workflow*

*and separate workflow branches therein* (See Figure 3-5, p. 63), *and wherein the workflow*

*definition further includes a construct to terminate the parallel processing of the workflow*

*when certain conditions are met* (See p. 12, Requirement 2, see p. 40, see p. 196, see Figure 5-

14 and sec. 5.4.4.3, p. 141, etc); *and*

*means for creating a workflow program according to said workflow definition, including*

*means for the computer to read the source file and process the plurality of workflow*

*constructs to activate a workflow, including creating separate workflow processes*

*corresponding to the separate workflow branches* (See Figure 1, p. 5, Workflow Definition

Tool),

*means for activating each of the separate workflow processes to subsequently generate*

*activities at the computer as defined by each workflow branch* (Figure 5-11, p. 130), *and means for determining when the certain conditions specified in the source file have occurred and then terminating the parallel processing of the workflow* (See Figure 5-14 and sec. 5.4.4.3).

Muller address a generic application program, Muller does not explicitly address, *program source file includes a source code and classes therein, and workflow definition created using a workflow language that is specified in the form of annotations to the source code and classes*

Peltz discloses “workflow definition is invoked using WSFL (See its “early work”, p. 5), where WSFL includes Java (See its first paragraph, p. 11), i.e. program source file that includes source code and classes therein, and JavaDoc annotation tags (See its p. 10) in the workflow language that is specified in the form of annotations to the source code and classes.

It is obvious to the ordinary in the art at the time to combine or to include program source file as Java Application because Java is commonly used by every developer for developing web and business applications, where it is an object-oriented program having source code and classes therein. Furthermore, using Java Application as of Peltz in the Muller’ Applications only yields the same and predictable results, and it is obvious for every ordinary in the art to include it in the teaching of Muller.

As per claim 51: Incorporated to the rejection of Claim 50, with regard to *The system of claim 50, wherein the workflow definition is invoked by a executing a software application* (See

Art Unit: 2191

Peltz: “workflow definition is invoked using WSFL” (See its “early work”, p. 5), where WSFL includes Java (See its first paragraph, p. 11), or JavaDoc annotation tags (See its p. 10)).

As per claim 52: Incorporated to the rejection of Claim 50, with regard to The *system of claim 50, wherein the plurality of workflow definition constructs are provided as XML commands* (Muller: See p. 314, 5. rule definitions in the source file include XML –Integration) *that are then used as annotations to the source code and classes* (See Peltz: p. 10).

Peltz discloses workflow definition files are used for web service (XML-Based Language), and constructs in the files are XML-commands (See its listings, e.g. p. 7, p.8, etc).

It is obvious to the ordinary in the art at the time to combine or provide XML-commands because XML is an open source used for Web-service. Furthermore, using XML-commands in a workflow definition file yields the same and predictable results, and it is obvious for every ordinary in the art to include it in the teaching of Muller for web purposes.

As per claim 53: With regards to, *The system of claim 50, further comprising a light-weight virtual machine at the computer that processes the workflow and that is enabled to, at a particular point in the workflow process, save the workflow's execution space including program stack and variable state, and, at a later point in time, revive the workflow at the same point in the workflow process using the saved program stack and variable state.*

It should be notice that virtual machine is only a browser using web-service. The Reference of Peltz discloses such of web service. It should be note that Java is a stack-based application program developed by Sun Microsystems,



Therefore, it is obvious to ordinary in the art for knowing that the further claim merely recites the rules, the techniques, and the principles, which have been already developed by others as requirement in the Web, and included in the process of workflows as of Muller or Peltz for conforming to the requirements of the Web.

As per claim 54: With regards to,

***The system of claim 50, wherein the program source file is a Web Service file that includes the workflow definition constructs.*** Refer to Peltz: The program with workflow definition constructs disclosed in Peltz is Java Web Service file.

As per claim 55: With regards to, ***The system of claim 54, wherein the workflow definition constructs of the Web Service file also references Java methods and variables for a software application running on the system and using the workflow.***

The workflows disclosed in Peltz is Java Web Service file using method and variables running on a computer system using workflow – For example see Figure 5, p. 10.

As per claim 56: With regards to, ***The system of claim 54, wherein workflow definitions are provided as a separate Work Flow file that includes workflow definition commands, and that are invoked by the Web Service file using the workflow definition constructs, to use the workflow as defined by the Work Flow file.***

The Workflow definition(s) in Muller figure 1-2 is provided as a separate Work Flow file that includes workflow definition commands, and that are invoked by the Applications program(s) using the workflow definition constructs in the files, to use the workflow as defined by the Application Program(s) as seen in Muller Figure 1-2,

Art Unit: 2191

It is obvious to the ordinary in the art to include separate Java Work Flow file as seen in the reference of Peltz because it yields predictable results.

As per Claim 71: Regarding, ***The system of claim 55, wherein the Web Service file includes the workflow definition constructs as a plurality of XML workflow annotations to the source code and classes defined in the Web Service file.*** See Peltz: p. 10.

As per Claim 72: Regarding, ***The system of claim 71, wherein the XML workflow annotations to the source code and classes define a flow logic that can then reference the methods and variables defined in the Web Service file.*** See Peltz: p. 10.

As Per Claims 57-63, and 73-74: The rejection of the claims is the same as of Claims 50-56, and 71-72. Refer to the rationale as addressed in the rejection of Claims 51-56, and 71-72 above.

As Per Claim 75: The rejection of the claim is the same as of Claim 50. Refer to the rationale as addressed in the rejection of Claim 50 above.

### ***Conclusion***

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ted T. Vo whose telephone number is (571) 272-3706. The examiner can normally be reached on 8:00AM to 4:30PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708.

Art Unit: 2191

The facsimile number for the organization where this application or proceeding is assigned is the Central Facsimile number **571-273-8300**.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

TTV  
May 22, 2008

/Ted T. Vo/  
Primary Examiner, Art Unit 2191